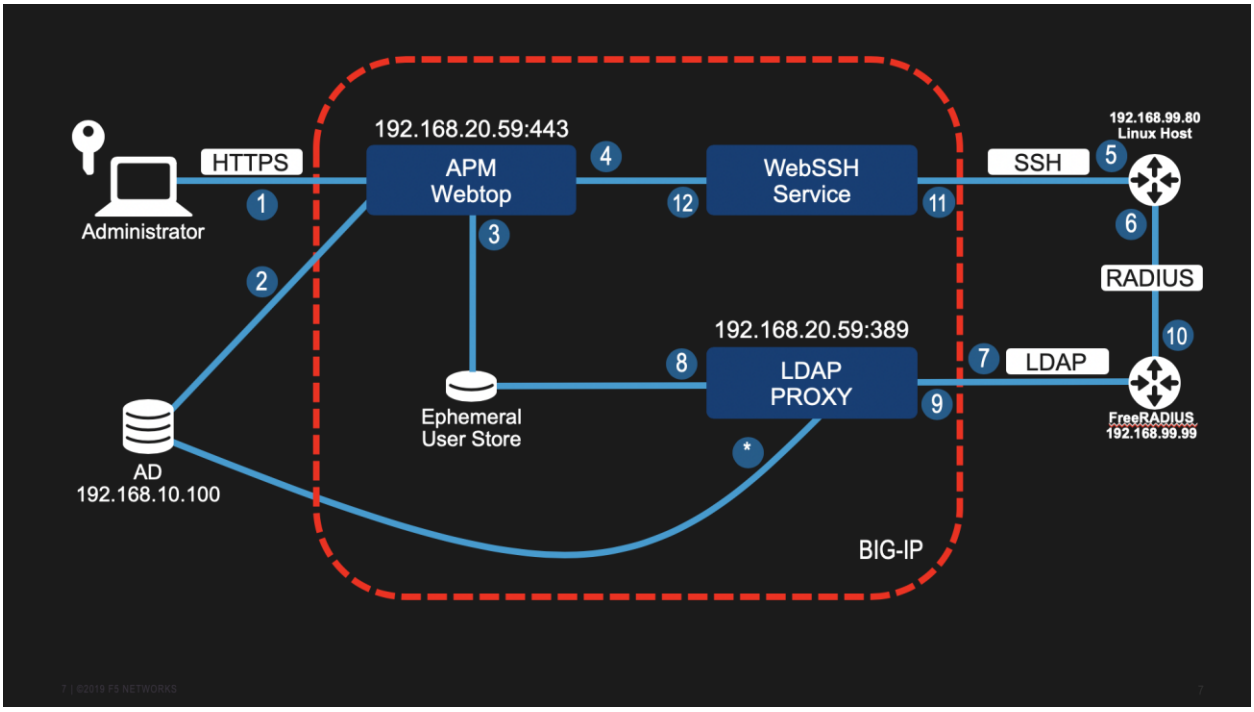
FreeRADIUS with LDAP authentication to Ephemeral Auth

Reference Configuration

Assumptions

- FreeRADIUS host is Ubuntu 18.04.3 LTS (192.168.99.99)
- RADIUS Client is Linux to FreeRADIUS server (192.168.99.80)
- Working PUA/Ephemeral Auth Environment (192.168.20.59)
- Working LDAP / Active Directory service proxied by BIG-IP LDAP Proxy (192.168.10.100)
- LDAP Service Account is “CN=ldap proxy,CN=Users,DC=mydomain,DC=local”



Configuration Steps

FreeRADIUS Server

Based on FreeRADIUS v3.x

Ubuntu 18.04.3 LTS Update

1. apt-get update
2. apt-get dist-upgrade

FreeRADIUS Installation

1. apt-get install freeradius

2. apt-get install freeradius-ldap

FreeRADIUS Configuration

Full FreeRADIUS configuration is out of scope for this document, however this solution makes use of the "rlm_ldap" module to authenticate a user against an LDAP database. More details on the configuration options for"rlm_ldap" can be found here: https://wiki.freeradius.org/modules/Rlm_ldap.

Before running FreeRADIUS, several files will need to be modified depending on the environment. Below are the relevant sections of each file to be modified and does not represent an entire configuration.

LDAP must first be enabled, this can be done by making a symbolic link to */etc/freeradius/3.0/mods_available/ldap* to */etc/freeradius/3.0/mods_enabled/ldap.*

/etc/freeradius/3.0/mods_enabled/ldap

Among the site-specific settings below, It's important to ensure the "chase_referrals" is set to "no". Otherwise LDAP requests may be handled by servers outside of the BIG-IP and result in a non-functional solution.

```
      server = '192.168.20.59'
      base_dn = 'DC=mydomain,DC=local'
…
      #  Administrator account for searching and possibly modifying.
      #  If using SASL + KRB5 these should be commented out.
      identity = 'CN=ldap proxy,CN=Users,DC=mydomain,DC=local'
      password = YourPassword
…
      #  Set to yes if you want to bind as the user after retrieving the
      #  Cleartext-Password. This will consume the login grace, and
      #  verify user authorization.
      edir_autz = yes

…
      user {
            #  Where to start searching in the tree for users
            base_dn = "${..base_dn}"

            #  Filter for user objects, should be specific enough
            #  to identify a single user object.
            #
            #  For Active Directory, you should use
            #  "samaccountname=" instead of "uid="
            #
            filter = "(samaccountname=%{%{Stripped-User-Name}:-%{User-Name}})"

…
      options {
            #  Control under which situations aliases are followed.
            #  May be one of 'never', 'searching', 'finding' or 'always'
…
```

```
        #  The following two configuration items control whether the
        #  server follows references returned by LDAP directory.
        #  They are  mostly for Active Directory compatibility.
        #  If you set these to 'no', then searches will likely return
        #  'operations error', instead of a useful result.
        #
        chase_referrals = no
```

etc/freeradius/3.0/mods-config/files/authorize

```
DEFAULT       Auth-Type := LDAP
        Fall-Through = 1
```

/etc/freeradius/3.0/clients

Add entry for your RADIUS client(s):

```
client ssh-host-1 {
    ipaddr = 192.168.99.80
    secret = radius_secret
    limit {
        max_connections = 50
        lifetime = 0
        idle_timeout = 30
    }
}
```

/etc/freeradius/3.0/sites-available/default

```
authenticate {
    #
…
    #  Uncomment it if you want to use ldap for authentication
    #
    #  Note that this means "check plain-text password against
    #  the ldap database", which means that EAP won't work,
    #  as it does not supply a plain-text password.
    #
    #  We do NOT recommend using this.  LDAP servers are databases.
    #  They are NOT authentication servers.  FreeRADIUS is an
    #  authentication server, and knows what to do with authentication.
    #  LDAP servers do not.
    #
    Auth-Type LDAP {
        ldap
    }
```

Testing

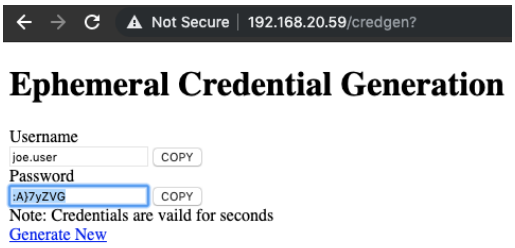
On the FreeRADIUS host, stop the *freeradius* service if already running:

```
root@freeradius:~# service freeradius stop
```

Run *freeradius* in debug mode in the foreground:

```
root@freeradius:~# freeradius -X
FreeRADIUS Version 3.0.16
Copyright (C) 1999-2017 The FreeRADIUS server project and contributors
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE
…
Listening on auth address 127.0.0.1 port 18120 bound to server inner-tunnel
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on proxy address * port 33654
Listening on proxy address :: port 41703
Ready to process requests
```

Connect an additional SSH terminal to the FreeRADIUS host to use the *radtest* utility. Log into the BIG-IP Webtop and generate a credential using *credgen.*



Use those credentials with the *radtest* utility. It may help to use sinlge quotes around the password to prevent special characters from being intercepted by the shell. Note that the RADIUS secret ‘testing123’ is the default one used for the ‘localhost’ client. You can verify this by looking for the *secret* option under ‘client localhost’ in ‘/etc/freeradius/3.0/clients’

```
root@freeradius:/etc/freeradius/3.0# radtest -t pap joe.user ':A}7yZVG' localhost 1 testing123
Sent Access-Request Id 45 from 0.0.0.0:47104 to 127.0.0.1:1812 length 78
        User-Name = "joe.user"
        User-Password = ":A}7yZVG"
        NAS-IP-Address = 192.168.99.99
        NAS-Port = 1
        Message-Authenticator = 0x00
        Cleartext-Password = ":A}7yZVG"
Received Access-Accept Id 45 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
```

We should see an “Access-Accept” for this test. This tells us our LDAP configuration is successful and we can move to configuring the RADIUS client (ssh-host-1) to use FreeRADIUS as it’s authentication server (which is now tiered to PUA / Ephemeral Auth).

```
(0) # Executing group from file /etc/freeradius/3.0/sites-enabled/default
(0)   Auth-Type LDAP {
rlm_ldap (ldap): Reserved connection (5)
(0) ldap: Login attempt by "joe.user"
(0) ldap: Using user DN from request "CN=joe user,CN=Users,DC=mydomain,DC=local"
(0) ldap: Waiting for bind result...
(0) ldap: Bind successful
(0) ldap: Bind as user "CN=joe user,CN=Users,DC=mydomain,DC=local" was successful
rlm_ldap (ldap): Released connection (5)
(0)     [ldap] = ok
(0)   } # Auth-Type LDAP = ok
(0) # Executing section post-auth from file /etc/freeradius/3.0/sites-enabled/default
(0)   post-auth {
(0)     update {
(0)       No attributes updated
(0)     } # update = noop
(0)     [exec] = noop
(0)     policy remove_reply_message_if_eap {
(0)       if (&reply:EAP-Message && &reply:Reply-Message) {
(0)       if (&reply:EAP-Message && &reply:Reply-Message)  -> FALSE
(0)       else {
(0)         [noop] = noop
(0)       } # else = noop
(0)     } # policy remove_reply_message_if_eap = noop
(0)   } # post-auth = noop
(0) Sent Access-Accept Id 45 from 127.0.0.1:1812 to 127.0.0.1:47104 length 0
(0) Finished request
Waking up in 4.9 seconds.
(0) Cleaning up request packet ID 45 with timestamp +174
Ready to process requests
```

Linux Server Configuration

ssh-host-1 is a Ubuntu 14.04.2 LTS server and will act as our RADIUS client. Detailed configuration of a Linux server for RADIUS authentication is out of scope for this document. This assumes that *libpam-radius-auth* is already installed.

/etc/pam_radius_auth.conf

```
192.168.99.99  radius_secret   1
```

/etc/pam.d/sshd

```
# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
auth       sufficient   pam_radius_auth.so
@include common-auth
```

We will be testing with “joe.user” make sure there is a “joe.user” account which already exists on this system:

```
# getent passwd |grep -i joe.user
joe.user:x:1001:1001:,,,:/home/joe.user:/bin/bash
```

If a user does not exist, one may be added with no password (this does not permit passwordless authentication, just that password comes from another source or via ssh private key)

```
root@ssh-host-1:~# adduser --disabled-password joe.user
Adding user `joe.user' ...
…
Enter the new value, or press ENTER for the default
            Full Name []:
…
Is the information correct? [Y/n]
root@ssh-host-1:~# getent passwd |grep -i joe.user
joe.user:x:1001:1001:,,,:/home/joe.user:/bin/bash
```

On the Linux host, run ***tail -f /var/log/auth.log***

As previously, using credgen generate a password for joe.user.

Open a new SSH session as joe.user to the Linux Host with the credentials generated by credgen.

At this point you should be able to use WebSSH to and PUA to log into the Linux Host through the FreeRADIUS server…